PCT

WELTORGANISATION FÜR GEISTIGES EIGENTUM
Internationales Büro

INTERNATIONALE ANMELDUNG VERÖFFENTLICHT NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT)

(51) Internationale Patentklassifikation 7 :
G06F 9/46, 12/14, 1/00

A1

(11) Internationale Veröffentlichungsnummer: **WO 00/11551**

(43) Internationales Veröffentlichungsdatum: 2. März 2000 (02.03.00)

(21) Internationales Aktenzeichen: PCT/DE99/02013

(22) Internationales Anmeldedatum: 1. Juli 1999 (01.07.99)

(30) Prioritätsdaten:
198 37 666.9 19. August 1998 (19.08.98) DE

(71) Anmelder *(für alle Bestimmungsstaaten ausser US)*: SIEMENS AKTIENGESELLSCHAFT [DE/DE]; Wittelsbacherplatz 2, D–80333 München (DE).

(72) Erfinder; und
(75) Erfinder/Anmelder *(nur für US)*: SCHÄFER, Manfred [DE/DE]; St.–Josef–Strasse 16, D–85661 Forstinning (DE).

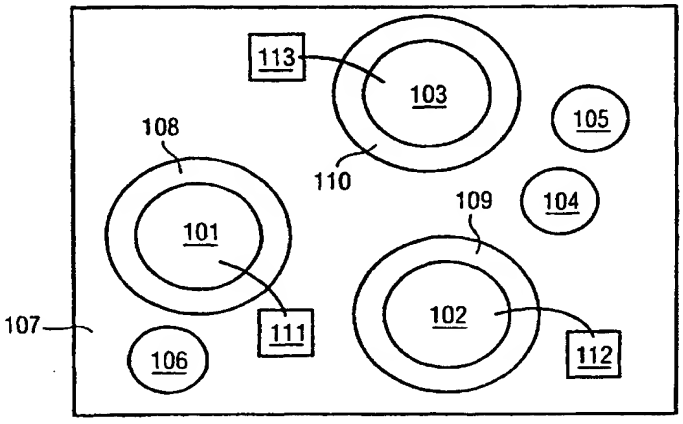
(74) Gemeinsamer Vertreter: SIEMENS AKTIENGESELLSCHAFT; Postfach 22 16 34, D–80506 München (DE).

(81) Bestimmungsstaaten: US, europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Veröffentlicht
Mit internationalem Recherchenbericht.
Vor Ablauf der für Änderungen der Ansprüche zugelassenen Frist; Veröffentlichung wird wiederholt falls Änderungen eintreffen.

(54) Title: METHOD, ARRAY AND SET OF SEVERAL ARRAYS FOR PROTECTING SEVERAL PROGRAMS AND/OR FILES FROM UNAUTHORIZED ACCESS BY A PROCESS

(54) Bezeichnung: VERFAHREN, ANORDNUNG SOWIE EIN SATZ MEHRERER ANORDNUNGEN ZUM SCHUTZ MEHRERER PROGRAMME UND/ODER MEHRERER DATEIEN VOR EINEM UNBEFUGTEN ZUGRIFF DURCH EINEN PROZESS



(57) Abstract

An area and a process file are assigned to each program to be protected. The process or processes that may run in the corresponding area are stored in a process file. When the program is running, a process attempting to access the program is checked to confirm whether the accessing process is included in the corresponding process file. The accessing process is only executed when it is included in the process file.

(57) **Zusammenfassung**

Es wird jedem zu schützenden Programm jeweils ein Bereich und eine Prozeß–Datei zugeordnet. In einer Prozeß–Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Bereich ablaufen darf oder dürfen. Während des Ablaufs des Programms wird für einen Prozeß, der auf das Programm zugreifen will, überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeßdatei angegeben ist. Der zugreifende Prozeß wird nur ausgeführt, wenn er in der Prozeßdatei angegeben ist.

Beschreibung

Verfahren, Anordnung sowie ein Satz mehrerer Anordnungen zum Schutz mehrerer Programme und/oder mehrerer Dateien vor einem unbefugten Zugriff durch einen Prozeß

Die Erfindung betrifft ein Verfahren, eine Anordnung sowie ein Satz mehrerer Anordnungen zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß.

Ein Verfahren und eine Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Anwender ist aus [1] oder [6] bekannt. Der Zugriffsschutz für ein Programm ist bei dem Verfahren aus [1] dadurch realisiert, daß jedem Benutzer eines Systems eine Zugriffsberechtigungsdatei zugeordnet wird. Versucht ein Prozeß, auf ein Programm zuzugreifen, so wird überprüft, ob der Benutzer, der den Prozeß gestartet hat, das Recht hat, auf das entsprechende Programm zuzugreifen. Der Zugriff wird nur gestattet, wenn der Prozeß von einem befugten und somit mit den Zugriffsrechten ausgestatteten Benutzer gestartet wurde.

Aus [2] ist ein sogenannter Virenscanner bekannt. Ein Virenscanner überprüft die gespeicherte, bekannte Folge von Daten, durch die das Programm realisiert ist. Wird eine Abweichung gegenüber der bekannten Folge festgestellt, so wird ein Benutzer des Systems benachrichtigt, daß möglicherweise das System mit einem Virus behaftet ist.

Aus [1] ist ferner ein Betriebssystem für einen Rechner bekannt. Das aus [1] bekannte Betriebssystem weist verschiedene Sicherheitslücken auf, durch die es einem Angreifer möglich ist, die Integrität von Programmen, die unter Verwendung des Betriebssystems durchgeführt werden, zu gefährden.

Ein möglicher Mechanismus, um den Schutz der Programme bei Verwendung dieses Betriebssystems zu gefährden ist ebenfalls in [5] beschrieben.

In [7] ist ein Computersystem zur Lizensierung von Software beschrieben.

Somit liegt der Erfindung das Problem zugrunde, mehrere Programme und/oder mehrere Dateien vor einem unbefugten Zugriff durch einen Prozeß zu schützen unter Verwendung eines Betriebssystems, welches grundsätzlich Sicherheitslücken aufweist.

Das Problem wird durch das Verfahren sowie durch die Anordnung gemäß den Merkmalen der unabhängigen Patentansprüche gelöst.

Bei einem Verfahren zum Schutz mehrerer Programme und/oder mehrerer Dateien vor einem unbefugten Zugriff durch einen Prozeß, ist jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils ein Adreßraum zugeordnet. Jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist ferner jeweils eine Prozeß-Datei zugeordnet, wobei in einer Prozeß-Datei gespeichert ist, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen. Während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will wird überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist. Für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, wird der zugreifende Prozeß gestartet; sonst wird der zugreifende Prozeß nicht gestartet.

Bei einem weiteren Verfahren zum Schutz mehrerer Programme und/oder mehrerer Dateien vor einem unbefugten Zugriff durch einen Prozeß, ist jedem zu schützenden Programm und/oder je-

der zu schützenden Datei jeweils ein Adreßraum zugeordnet. Jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist ferner jeweils eine Prozeß-Datei zugeordnet, wobei in einer Prozeß-Datei gespeichert ist, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen. Während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder einer zu schützenden Datei zugreifen will wird überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist. Für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, wird der zugreifende Prozeß weitergeführt; sonst wird der zugreifende Prozeß beendet.

Eine Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß, weist einen Prozessor auf, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:

- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,
- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei wird für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will, überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, wird der zugreifende Prozeß gestartet, und
- sonst wird der zugreifende Prozeß nicht gestartet.

Eine weitere Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß, weist einen Pro-

zessor auf, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,
- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei wird für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, wird der zugreifende Prozeß weitergeführt, und
- sonst wird der zugreifende Prozeß beendet.

Ein Satz mehrerer Anordnungen und eine mit jeder Anordnung des Satzes mehrerer Anordnungen verbundene Server-Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß ist derart eingerichtet, daß
jede Anordnung einen Prozessor aufweist, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,
- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei wird für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,

- für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, wird der zugreifende Prozeß gestartet oder weitergeführt, und
- sonst wird ein Alarmsignal generiert und an die Server-Anordnung gesendet,

und die Server-Anordnung einen Prozessor aufweist, der derart eingerichtet ist, daß abhängig von mindestens einem empfangenen Alarmsignal eine vorgegebene Aktion ausgelöst wird.

Unter Adreßraum ist in diesem Zusammenhang ein Programm-Bereich zu verstehen, der jeweils einem Programm zugeordnet ist.

Durch die Erfindung werden mehrere Sicherheitslücken des in [1] beschriebenen Betriebssystems geschlossen.

Weiterhin wird das jeweils zu schützende Programm gegen einen prozeduralen Angriff (Angriff auf einen Prozeß), z.B. gegen ein Trojanisches Pferd, geschützt.

Ferner ist ein erheblicher Vorteil der Erfindung darin zu sehen, daß bei skalierbarem Aufwand ein definiertes Maß an Sicherheit für das zu schützende Programm gewährleistet werden kann.

Durch den Satz mehrerer Anordnungen, die jeweils mit der Server-Anordnung verbunden sind, ist ein Schutz lokal bei den Anordnungen möglich derart, daß bei einem erkannten Angriff ein Alarmsignal generiert und an die Server-Anordnung gesendet wird, in der zentral eine vorgegebene Aktion ausgeführt wird. Auf diese Weise ist die Entdeckung lokaler Prozesse möglich, die der Server-Anordnung selbst nicht bekannt sind.

Bevorzugte Weiterbildungen der Erfindung ergeben sich aus den abhängigen Ansprüchen.

Es ist in einer Weiterbildung zur Erhöhung des erreichbaren Sicherheitsniveaus vorteilhaft, zumindest einem Teil der in einer Prozeß-Datei angegebenen Prozesse einen eindeutig kennzeichnenden kryptographischen Wert zu bilden, wobei der jeweilige Wert in der Prozeß-Datei enthalten ist. Für den zugreifenden Prozeß wird dessen kryptographischer Wert gebildet und bei der Überprüfung werden die kryptographischen Werte der Prozesse miteinander verglichen.

Der kryptographische Wert kann eine digitale Signatur sein. Er kann aber auch unter Verwendung einer Hash-Funktion, allgemein einer Einwegfunktion, gebildet werden.

In einer weiteren Ausgestaltung ist es vorteilhaft, in einem Aufrufmechanismus für eine Funktion eines Betriebssystemkerns, mit dem die Programme ausgeführt werden, einen Aufruf des zugreifenden Prozesses zu einer Überprüfungsfunktion weiterzuleiten, in der die Überprüfung erfolgt. Auf diese Weise ist eine effiziente und somit kostengünstige Realisierung der Erfindung möglich.

Die Überprüfungsfunktion kann als dynamisch bindbare Datei in den überwachten Adreßraum eingebunden werden, wodurch eine weitere Verbesserung in der Schutzwirkung erreicht wird.

Ein Aufruf eines zugreifenden Prozesses kann auch zu einer Überprüfungsfunktion, die in dem Betriebssystemkern integriert ist, weitergeleitet werden, wobei die Überprüfung in der Überprüfungsfunktion erfolgt. Auf diese Weise kann die erreichbare Sicherheit des Schutzes für die Programme noch weiter erhöht werden.

Eine weitere Erhöhung des erreichbaren Sicherheitsniveaus kann gewährleistet werden, wenn ein Schutzprogramm, welches derart eingerichtet ist, daß die Erfindung ausführbar ist, verschlüsselt gespeichert ist und zu Beginn des Verfahrens entschlüsselt wird. Nach der Entschlüsselung des Schutzpro-

gramms kann dessen Integrität überprüft werden und das Verfahren wird nur dann ausgeführt, wenn die Integrität des Schutzprogrammes gewährleistet ist. Nach der Integritätsprüfung des Schutzprogramms kann die Integrität aller in den Prozeß-Dateien enthaltenen Prozesse überprüft werden und das Verfahren wird nur ausgeführt, wenn die Integrität des Schutzprogramms gewährleistet ist. Nach der Integritätsprüfung der Prozesse, kann die Integrität des zu schützenden Programms überprüft werden und das Programm sollte nur ausgeführt werden, wenn die Integrität des Schutzprogramms gewährleistet ist.

Die Erfindung ist vorteilhaft einsetzbar in dem in [1] beschriebenen Betriebssystem.

Obwohl das im weiteren erläuterte Ausführungsbeispiel den Schutz von Programmen beschreibt, so ist ebenfalls ein Schutz mehrerer Dateien ohne weiteres gemäß der gleichen Vorgehensweise möglich.

Ein Ausführungsbeispiel der Erfindung ist in den Figuren dargesellt und wird im weiteren näher erläutert.

Es zeigen

Figur 1 eine Skizze, in der das der Erfindung zugrundeliegende Prinzip symbolisch dargestellt ist,

Figur 2 eine Skizze, in der ein Prozeßschichtenmodell dargestellt ist;

Figur 3 ein Blockdiagramm, in dem ein Rechnernetz dargestellt ist;

Figur 4 ein Ablaufdiagramm, in dem die einzelnen Verfahrensschritte des Ausführungsbeispiels dargestellt sind;

Figur 5 eine Skizze, in der das Prinzip einer möglichen Integration der Erfindung in einem Betriebssystem dargestellt ist;

Figur 6 eine Skizze, in der die mögliche Realisierung gemäß Figur 5 detailliert dargestellt ist;

Figur 7 eine Skizze, in der eine weitere mögliche Integration der Erfindung in ein Betriebssystem dargestellt ist.

Fig.3 zeigt einen ersten Rechner 301 mit einer Eingangs-/Ausgangsschnittstelle 302, die über einen Bus 303 mit einem Speicher 304 und einem Prozessor 305 verbunden ist. Über die Eingangs-/Ausgangsschnittstelle 302 ist der erste Rechner 301 über ein Rechnernetz 306 mit einer Vielzahl von Rechnern 307, 308, 309, 310, 311 verbunden.

Das zur Übertragung digitaler Daten verwendete Kommunikationsprotokoll ist das TCP/IP-Protokoll (Transmission Control Protocol/Internet Protocol).

Die Erfindung schützt den ersten Rechner 301 vor unbefugten Zugriffen von Prozessen, die entweder in dem Speicher 304 des ersten Rechners 301 gespeichert sind oder von den weiteren Rechnern 307, 308, 309, 310, 311, die auf den ersten Rechner 301 zugreifen/einwirken.

In dem ersten Rechner 301 ist das in [1] beschriebene Betriebssystem implementiert.

Das im weiteren detailliert erläuterte Prinzip, welches dem Verfahren bzw. der Anordnung zugrunde liegt, ist anschaulich in Fig.1 dargestellt.

Symbolisch dargestellte Programme 101, 102, 103 sollen durch die Erfindung gegen unbefugten Zugriff durch mindestens einen

Prozeß 104, 105, 106, der oder die auf ein Programm 101, 102, 103 zugreifen wollen, geschützt werden.

Die Programme 101, 102, 103 und Prozesse 104, 105, 106 verwenden als Betriebssystem, dargestellt als eine die Programme 101, 102, 103 und Prozesse 104, 105, 106 umrahmende Einheit 107 das in [1] beschriebene Betriebssystem.

Durch das Verfahren bzw. durch die Anordnung wird anschaulich um jedes zu schützende Programm 101, 102, 103 eine programmspezifische „Schutzhülle" 108, 109, 110 gebildet. Durch die programmspezifische Sicherung wird ein frei skalierbares Sicherheitsniveau für das zu schützende Programm 101, 102, 103 erreicht.

Das Verfahren bzw. die Anordnung kann, wie in einem Prozeßschichtenmodell 201 in **Fig.2** dargestellt, auf verschiedenen logischen Ebenen zu schützender Programme 101, 102, 103 realisiert werden. **Fig.2** zeigt drei logische Ebenen in dem Prozeßschichtenmodell 201.

Die Sicherung vor einem Angreifer 205 kann auf der Ebene zu schützender Anwendungsprogramme 202, auf der Ebene zu schützender Betriebssystemprogramme 203 sowie auf der Ebene des Betriebssystemkerns, und auf der System-Hardware 207 erfolgen.

Je näher an der System-Hardware 207 die Sicherung eines Programms erfolgt, desto größer ist das Sicherheitsniveau, das durch die Erfindung erreicht wird.

Die Schutzhülle 206 wird zur Laufzeit des zu schützenden Programms 101, 102, 103 um das Programm „gelegt".

Das Verfahren wird in Form zyklischer, nebenläufiger Prozesse realisiert. Das Verfahren wird anhand des Ablaufdiagramms, das in **Fig.4** dargestellt ist, erläutert.

Als erstes nach dem Start des Betriebssystems (Schritt 401) wird ein Schutzprogramm gestartet (Schritt 402). Das Schutzprogramm ist derart eingerichtet, daß das im weiteren beschriebene Verfahren ausführbar ist. Das Schutzprogramm ist in verschlüsselter Form gespeichert, wodurch eine Veränderung des Schutzprogramms selbst nicht möglich ist.

Auch wird durch die Verschlüsselung des Schutzprogramms eine detaillierte Analyse des das Verfahren ausführenden Programms verhindert.

Zum Start des Schutzprogramms (Schritt 402) wird das Schutzprogramm durch eine vorgegebene Startroutine, die den zur Entschlüsselung des Schutzprogramms erforderlichen Schlüssel und eventuell weitere Grundfunktionen des Betriebssystems enthält, entschlüsselt, wodurch der eigentliche Programmcode des Schutzprogramms ermittelt wird.

Auf diese Weise ist das Schutzprogramm im aktiven Zustand vor off-line Angriffen wie disassemblieren, debuggen, patchen, ..., geschützt.

Nach der Entschlüsselung (Schritt 402) des Schutzprogramms wird die Integrität des Schutzprogramms dynamisch überprüft (Schritt 403).

Ist die Integrität des Schutzprogramms nicht gewährleistet, so wird das Verfahren abgebrochen (Schritt 404).

In einem weiteren Schritt wird die Integrität der Prozesse des Betriebssystems dynamisch überprüft (Schritt 405). Bei negativer Integritätsprüfung wird das Verfahren wiederum abgebrochen (Schritt 404).

Ist die Integrität der Prozesse des Betriebssystems 405 gewährleistet, so wird das zu schützende Programm gestartet.

Das oben beschriebene Verfahren wird für jedes zu schützende Programm 101, 102, 103, dynamisch durchgeführt.

Jedem zu schützenden Progamm 101, 102, 103 ist jeweils eine Prozeßdatei 111, 112, 113 zugeordnet.

In einer Prozeßdatei 111, 112, 113 ist für das zu schützende Programm 101, 102, 103, dem die Prozeß-Datei 111, 112, 113 zugeordnet ist, angegeben, welche Prozesse in einem Adreßraum, der ebenfalls jedem Programm 101, 102, 103 eindeutig zugeordnet ist, ablaufen dürfen. In den Prozeß-Dateien sind die Angaben mittels einer den jeweiligen Prozeß eindeutig kennzeichnenden Hash-Funktion gespeichert.

Nach Start des jeweiligen Programms 101, 102, 103 (Schritt 407) wird die Integrität des Programms 101, 102, 103 selbst überprüft (Schritt 408).

Bei negativer Integritätsprüfung wird wiederum das Verfahren abgebrochen (Schritt 404).

Bei positiver Integritätsprüfung, d.h. wenn die Integrität des Programms 101, 102, 103 gewährleistet ist, wird das Verfahren für das zu schützende Programm 101, 102, 103 solange wiederholt, bis das zu schützende Programm selbst beendet wird (Schritt 409).

Das Verfahren wird abhängig von einem vorgebbaren Ereignis oder in einem vorgebbaren Zeitabstand zwischen zwei Ausführungen des Verfahrens iteriert (Schritt 410).

Sobald ein Prozeß 104, 105, 106 auf den Adreßraum bzw. das Programm 101, 102, 103 selbst zugreifen will, wird in einem weiteren Schritt (Schritt 411) überprüft, ob der Prozeß 104, 105, 106 in der Prozeß-Datei des zu schützenden Programms

101, 102, 103, auf das der Prozeß 104, 105, 106 zugreifen will, enthalten ist oder nicht.

Dies erfolgt durch Bildung eines Hash-Wertes über den zugreifenden Prozeß 104, 105, 106 und Vergleich des Hash-Wertes des zugreifenden Prozesses 104, 105, 106 mit den Hash-Werten, die in der Prozeß-Datei gespeichert sind. Ist der zugreifende Prozeß in der Prozeß-Datei des zu schützenden Programms 101, 102, 103 angegeben, so wird der Prozeß 104, 105, 106 durchgeführt (Schritt 412).

Sonst wird der Prozeß 104, 105, 106 nicht gestartet (Schritt 413) und der Benutzer wird über einen möglichen Angriff auf sein Programm 101, 102, 103 informiert.

In vorgebbaren Zeitabständen oder ereignisgesteuert wird für jeden in dem Adreßraum eines Programms 101, 102, 103 aktiven Prozeß, d.h. ablaufenden Prozeß, überprüft, ob der jeweilige Prozeß in der Prozeß-Datei des entsprechenden Programms 101, 102, 103, dessen Adreßraum untersucht wird, enthalten ist. Ist dies nicht der Fall, so wird der entsprechende Prozeß beendet und der Benutzer wird auf einen möglichen Angriff auf sein Programm 101, 102, 103 hingewiesen.

Auf diese Weise ist eine regelmäßige Überwachung des Programms gewährleistet.

Im weiteren werden Möglichkeiten zur Integration des oben beschriebenen Verfahrens in das in [3] beschriebene Betriebssystem dargestellt:

1. Möglichkeit:

Integration einer dynamisch bindbaren Datei 501 in die Anwendungs-Programmier-Schnittstelle (Application Programming Interface, API) (vgl. **Fig.5**).

Die dynamisch bindbare Datei 501 wird im Adreßraum des potentiellen Angreifer-Programms 502 aktiv. Unter Verwendung der dynamisch bindbaren Datei 501 werden folgende Schritte in dem Adreßraum des potentiellen Angreifer-Programms 502 ausgeführt:

- Von der dynamisch bindbaren Datei 501 werden alle Kennzeichnungen (Modulhändel) von jeder weiteren dynamisch bindbaren Datei ermittelt, die zu überwachende Schnittstellen-Aufrufe enthält. Damit können alle Zugriffe auf alle zu überwachenden dynamisch bindbaren Dateien ermöglicht werden.
  Als für Angriffe relevant werden alle Schnittstellen-Aufrufe zum direkten oder indirekten Starten, Beenden und Kontrollieren von Prozessen, z.B. schreibende Zugriffe auf den Prozeßspeicher, Änderungen der Zugriffsrechte, prinzipiell alle Schnittstellen-Befehle, die mit fremden Prozeßmarkierungen (Prozeßhandles) arbeiten, alle Befehle zum Realisieren von Message Hooks (eine programmierbare Filterfunktion für Nachrichten zur GUI-Interkommunikation (Graphic User Interface) und zur Prozeßinterkommunikation), und für Debugging-Zwecke betrachtet.
  Unter dem Ausdruck „indirektes Starten" werden auch Mechanismen eines Compilers zur Selbstmodifikation von Programmcode, OLE (Object Link Embedding) und RPC-Mechanismen (Remote Procedure Call) sowie Zugriffe aus anderen Betriebssystem-Programmier-Schnittstellen verstanden. Ferner umfaßt dieser Ausdruck auch die Kontrolle der Mechanismen, die für den Ablauf dynamisch bindbarer Dateien 501 (ActiveX-Befehle usw.) eingesetzt werden.
- Ferner werden der dynamischen bindbaren Datei 501 Schreibrechte für den Adreßraum zugeordnet.
- Anstelle des Original-Schnittstellen-Aufrufs wird ein Sprung-Befehl gespeichert und die ersetzten Befehle des Original-Codes 503 werden gesichert.

Versucht nun ein Prozeß 504 über einen Schnittstellenaufruf APIFktCall() auf das zu schützende Programm 502 zuzugreifen

bzw. ein bisher inaktives Programm zu starten, so wird durch die dynamisch bindbare Datei 501 das Schutzprogramm 506 aufgerufen. Als Übergabeparameter für das Schutzprogramm 506 wird dem Schutzprogramm 506 angegeben, welches Programm 502 durch den zugreifenden Prozeß 504 aufgerufen werden soll.

Durch einen Vergleich mit in der Prozeß-Datei des jeweiligen Programms 502 angegebenen zugelassenen Schnittstellen-Aufrufen wird dann gemäß dem oben beschriebenen Verfahren entschieden, ob der Schnittstellen-Aufruf 505 zugelassen wird oder nicht.

Der zugreifende Prozeß 504 wird dann entweder ausgeführt oder „abgeblockt". Wird in dem Schutzprogramm 506 entschieden, daß der zugreifende Prozeß 504 ausgeführt werden soll, so wird der Original-Code des Schnittstellen-Aufrufs 503 ausgeführt und nach dessen Ausführung wird ein Return-Code 507 an den zugreifenden Prozeß 504 zurückgegeben. Sonst wird eine Fehlermeldung 508 an den zugreifenden Prozeß 504 gemeldet.

Das oben beschriebene Verfahren verwendet den grundsätzlichen Mechanismus der sogenannten DLL-Injektion, die aus [5] bekannt ist.

2. Möglichkeit:

Fig.6 zeigt das in **Fig.5** beschriebene Prinzip in verfeinerter Realisierung. Diese Variante eignet sich insbesondere für den Fall, daß sich fälschlicherweise ein als „sicher" deklarierter Prozeß in dem Adreßraum befindet, der die dynamisch bindbare Datei 601 selbst angreift.

Durch das im folgenden beschriebene Verfahren wird gewährleistet, daß ein Zugriff auf vorgegebene Daten nur aus der dynamisch bindbaren Datei 601 selbst heraus möglich ist und Zugriffe aus einem anderen Befehlsfolgesegment, insbesondere einer angreifenden Applikation, verhindert wird.

Das im folgenden dargestellte Verfahren wird vorzugsweise mit statischer Modifikation des oben beschriebenen Verfahrens realisiert.

Für das im weiteren beschriebene Verfahren werden folgende Annahmen getroffen:

- Zu schützende Daten werden in einem geschützten Bereich 602 gespeichert, die bei der Initialisierung der dynamisch bindbaren Datei 601 angelegt und beschrieben wird und anschließend ein Schutzattribut erhält, so daß auf den geschützten Bereich 602 nur durch die dynamisch bindbare Datei 601 selbst zugegriffen werden kann.
- Alle Bereiche, die ausführbaren Code enthalten, erhalten das Schutzattribut „page_execute", wodurch verhindert wird, daß der entsprechende ausführbare Code nicht verändert werden kann, ohne daß das Schutzattribut zuvor geändert wird.
- Jede Schnittstellen-Funktion 603 wird auf folgende Weise gesichert: Eine Einsprungsadresse für die Schnittstellen-Funktion 603 wird durch eine modifizierte Einsprungsadresse, die zu einer modifizierten Schnittstellen-Funktion 604 führt, ersetzt.
  In der modifizierten Schnittstellen-Funktion wird verzweigt zu einem Schnittstellen-Prozeß 605, der in der dynamisch bindbaren Datei 601 enthalten ist. Dieser Schnittstellen-Prozeß verzweigt zu dem Schutzprogramm 606, durch das für einen aufrufenden Prozeß 607, der mit einem Schnittstellenaufruf 608 versucht, auf die Schnittstellenfunktion 603 zuzugreifen, überprüft, ob dieser Aufruf für den zugreifenden Prozeß 607 zugelassen ist.
  Ist dies der Fall, so wird die Schnittstellenfunktion 603 ausgeführt und es wird nach Durchführung der Schnittstellenfunktion 603 wiederum in die modifizierte Schnittstellenfunktion 604 verzweigt, was durch einen Pfeil 609 symbolisiert ist. Nach Ausführung weiterer vorgebbarer Befehle wird in eine Schnittstellen-Rückkehr-Funktion 610 verzweigt, was durch einen Pfeil 611 angedeutet ist. Dies er-

folgt durch einen Sprungbefehl. In der Schnittstellen-Rückkehr-Funktion 610 wird noch einmal überprüft (Schritt 612), ob der Schnittstellen-Aufruf 608 zugelassen ist. Wenn dies nicht der Fall ist, wird eine Fehlermeldung 613 an den Prozeß 607 gesendet. Dies erfolgt ebenso, wenn in der Schnittstellen-Funktion 605 unter Verwendung des oben beschriebenen Verfahrens ermittelt wurde, daß der Schnittstellen-Aufruf nicht zugelassen ist.
Wird jedoch auch in dem Überprüfungsschritt 612 in der Schnittstellen-Rückkehr-Funktion 610 ermittelt, daß der Schnittstellen-Aufruf zugelassen ist, so wird das Ergebnis der aufgerufenen Schnittstellen-Funktion an den zugreifenden Prozeß 607 gesendet (Schritt 614).

3. Möglichkeit:

Die Erfindung kann auch in dem Betriebssystemkern integriert werden. Durch diese Ausführungsform wird erreicht, daß ein Kontrollmechanismus integrierbar ist, dessen Umgehung unter Benutzerzugriffsrechten nicht mehr möglich ist, sondern nur unter den Zugriffsrechten des Systemadministrators. Damit wird das erreichbare Sicherheitsniveau erheblich gesteigert.

Ein hierzu verwendbarer Integrationsmechanismus ist in [4] beschrieben. Bei diesem Mechanismus werden Schnittstellen-Aufrufe beim Übertritt in den Modus des Betriebssystemkerns (Kernel-Modus) im Betriebssystemkern selbst "abgefangen". Der Verwaltungsmechanismus für die Unterbrechnungs-Routine ist in diesem Fall im Betriebssystemkern realisiert und ist daher gegen Zugriffe von Prozessen, die im Benutzermodus aktiv sind, geschützt. Eine Übersicht verschiedener solcher möglichen alternativen Implementierungsmöglichkeiten ist in [4] zu finden.

Fig.7 zeigt in einer Übersicht zwei oben beschriebene Möglichkeiten zur Realisierung.

Ein Anwendungsprogramm 701 (Applikation) verwendet zum Programmablauf Funktionen des Betriebssystems.

Die Funktionen des Betriebssystems sind gruppiert in Funktionen 702 des Betriebssystems in einem Benutzermodus (User-Mode) und in Funktionen 703 des Kernel-Modus. Die Funktionen sind symbolisch jeweils als Blöcke dargestellt.

Durch dynamisch bindbare Dateien (*.dll) ist es möglich, das Verfahren im Rahmen des Benutzermodus zu integrieren, was durch einen ersten Block 704 unter Verwendung einer dynamisch bindbaren Datei "NTDLL.DLL" 705 des in [1] beschriebenen Betriebssystems erfolgen kann.

Die weitere Integrationsmöglichkeit des Verfahrens in den Betriebssystemkern ist durch einen zweiten Block 706 angedeutet, wobei bei dieser Integrationsvariante der Mechanismus beim Übergang des Benutzermodus in den Kernel-Modus integriert wird.

Im weiteren werden einige Alternativen zu den oben erläuterten Ausführungsbeispielen dargestellt:

Der Schutz des Schutzprogramms kann dadurch erhöht werden, daß die Startroutine im Betriebssystemkern integriert ist, z.B. als sogenannter Kernel-Modus-Prozeß oder auch System Service.

Die dynamisch bindbare Datei kann auch sowohl statisch als auch dynamisch, d.h. lediglich während der Laufzeit des zu schützenden Programms 502 oder während der gesamten Laufzeit des Betriebssystems vorgesehen sein.

Auch können alternativ Software-Unterbrechungs-Routinen als Alternative zu der dynamisch bindbaren Datei verwendet werden.

Die Erfindung kann sowohl durch Software als auch durch Hardware oder zum Teil durch Software und zum Teil durch Hardware realisiert werden.

Ferner kann, wie in Fig.4 dargestellt, bei negativer Integritätsprüfung (Schritt 408) in einem weiteren Überprüfungsschritt (Schritt 414) überprüft werden, ob ein Nachladen des originalen Schutzprogramms und/oder eines zu schützenden Programms von einer vertrauenswürdigen Instanz möglich ist, oder ob eine Wiederherstellung des Schutzprogramms und/oder des zu schützenden Programms möglich ist, derart, daß dessen/deren Integrität gewährleistet ist.

Ist dies nicht möglich, so wird das Verfahren abgebrochen (Schritt 404).

Ist dies jedoch möglich, so wird die Integrität des neu geladenen bzw. wiederhergestellten Schutzprogramms weiter dynamisch überprüft (Schritt 403).

In einer weiteren Variante ist es vorgesehen, daß die Rechner mit einem Server, in Fig.1 der erste Rechner, verbunden sind.

In jedem Rechner wird das oben beschriebene Verfahren zum Schutz eines Programms und/oder einer Datei durchgeführt.

Für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei nicht angegeben ist, wird ein Alarmsignal generiert und an den Server gesendet. In dem Server wird abhängig von mindestens einem empfangenen Alarmsignal eine vorgegebene Aktion ausgelöst wird, beispielsweise ein zentral gesteuerter Abbruch eines Prozesses.

Im Rahmen dieses Dokuments wurden folgende Veröffentlichungen zitiert:

[1] Microsoft Windows NT workstation resource kit: comprehensive resource guide and utilities for Windows NT 4.0, ISBN 1-57231-343-9, Microsoft Press, C. Doyle (ed.), S. 117 - 118, 1996

[2] Dr. Solomon's Anti-Virus Toolkit Workstation, öffentlich zugänglich im Internet am 03.07.1998 unter der Internet-Adresse: http://www.drsolomon.de/produkte/avtkws/avtkws.html

[3] M. Petrek, Under the hood, MS-Journal, No. 12, December 1996,

[4] M. Russinovich, Windows NT System-Call Hooking, Dr. Dobb's Journal, S. 42 - 46, Januar 1997,

[5] J. Richter, Advanced Windows, ISBN 1-573231-548-2, 3rd Edition, S. 899 ff., Kapitel: Breaking Through Process-Boundary Walls, 1997

[6] US 5 390 310

[7] US 5 023 907

Patentansprüche

1. Verfahren zum Schutz mehrerer Programme und/oder mehrerer Dateien vor einem unbefugten Zugriff durch einen Prozeß,
- bei dem jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils ein Adreßraum zugeordnet ist,
- bei dem jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils eine Prozeß-Datei zugeordnet ist,
- bei dem in einer Prozeß-Datei gespeichert ist, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- bei dem während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft wird, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- bei dem für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, der zugreifende Prozeß gestartet wird, und
- sonst der zugreifende Prozeß nicht gestartet wird.

2. Verfahren zum Schutz mehrerer Programme und/oder mehrerer Dateien vor einem unbefugten Zugriff durch einen Prozeß,
- bei dem jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils ein Adreßraum zugeordnet ist,
- bei dem jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils eine Prozeß-Datei zugeordnet ist,
- bei dem in einer Prozeß-Datei gespeichert ist, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- bei dem während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder einer zu schützenden Datei zugreifen will überprüft wird, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,

- bei dem für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, der zugreifende Prozeß weitergeführt wird, und
- sonst der zugreifende Prozeß beendet wird.

3. Verfahren nach Anspruch 1 oder 2,
- bei dem für zumindest einen Teil der in einer Prozeß-Datei angegebenen Prozesse ein den Prozeß eindeutig kennzeichnender kryptographischer Wert gebildet wird,
- bei dem in der Prozeß-Datei jeweils der kryptographische Wert eines Prozesses enthalten ist,
- bei dem für den zugreifenden Prozeß dessen kryptographischer Wert gebildet wird, und
- bei dem bei der Überprüfung die kryptographischen Werte der Prozesse miteinander verglichen werden.

4. Verfahren nach einem der Ansprüche 1 bis 3,
bei dem in einem Aufrufmechanismus für eine Funktion eines Betriebssystemkerns, mit dem die Programme ausgeführt werden, ein Aufruf des zugreifenden Prozesses einer Überprüfungsfunktion zugeleitet wird, in der die Überprüfung erfolgt.

5. Verfahren nach Anspruch 4,
bei dem die Überprüfungsfunktion als dynamisch bindbare Datei in den Adreßraum eingebunden wird.

6. Verfahren nach einem der Ansprüche 1 bis 3,
- bei dem ein Aufruf des zugreifenden Prozesses einer Überprüfungsfunktion zugeleitet wird, in der die Überprüfung erfolgt, und
- bei dem die Überprüfungsfunktion in einen Betriebssystemkern eines Betriebssystems, mit dem die Programme ausgeführt werden, integriert ist.

7. Verfahren nach einem der Ansprüche 1 bis 6,
bei dem das Betriebssystem Windows NT ist.

8. Verfahren nach einem der Ansprüche 1 bis 7,
bei dem in einem vorgebbaren Zeitabstand für jeden aktiven Prozeß, der zusammen mit einem zu schützenden Programm und/oder einer zu schützenden Datei abläuft, überprüft wird, ob der aktive Prozeß in der Prozeß-Datei, die dem zu schützenden Programm und/oder der zu schützenden Datei zugeordnet ist, enthalten ist und der Prozeß beendet wird, falls dies nicht der Fall ist.

9. Verfahren nach einem der Ansprüche 1 bis 8,
bei dem abhängig von einem vorgebbaren Ereignis für jeden aktiven Prozeß, der zusammen mit einem zu schützenden Programm und/oder einer zu schützenden Datei abläuft, überprüft wird, ob der aktive Prozeß in der Prozeß-Datei, die dem zu schützenden Programm zugeordnet ist, enthalten ist und der Prozeß beendet wird, falls dies nicht der Fall ist.

10. Verfahren nach einem der Ansprüche 1 bis 9,
bei dem ein Schutzprogramm, welches derart eingerichtet ist, daß das Verfahren ausführbar ist, verschlüsselt gespeichert ist und zu Beginn des Verfahrens entschlüsselt wird

11. Verfahren nach einem der Ansprüche 1 bis 10,
bei dem die zu schützenden Programme und/oder die zu schützenden Dateien, verschlüsselt gespeichert sind und zu Beginn des Verfahrens entschlüsselt werden.

12. Verfahren nach Anspruch 10,
bei dem nach der Entschlüsselung des Schutzprogramms dessen Integrität überprüft wird und das Verfahren nur ausgeführt wird, wenn die Integrität des Schutzprogramms gewährleistet ist.

13. Verfahren nach Anspruch 12,
bei dem nach der Integritätsprüfung des Schutzprogramms die Integrität aller in den Prozeß-Dateien enthaltenen Prozesse

überprüft wird und das Verfahren nur ausgeführt wird, wenn die Integrität des Schutzprogramms gewährleistet ist.

14. Verfahren nach Anspruch 13,
bei dem nach der Integritätsprüfung der Prozesse die Integrität des zu schützenden Programms und/oder der zu schützenden Datei überprüft wird und das Verfahren nur ausgeführt wird, wenn die Integrität des Schutzprogramms gewährleistet ist.

15. Verfahren nach Anspruch 13 oder 14,
bei dem mindestens eine der Integritätsprüfung unter Verwendung eines kryptographischen Verfahrens erfolgt.

16. Verfahren nach einem der Ansprüche 3 bis 15,
bei dem der kryptographische Wert durch Anwendung einer Hash-Funktion gebildet wird

17. Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß,
mit einem Prozessor, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,
- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei wird für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, wird der zugreifende Prozeß gestartet, und
- sonst wird der zugreifende Prozeß nicht gestartet.

18. Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß,
mit einem Prozessor, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,
- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei wird für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, wird der zugreifende Prozeß weitergeführt, und
- sonst wird der zugreifende Prozeß beendet.

19. Anordnung nach Anspruch 17 oder 18,
bei der der Prozessor derart eingerichtet ist, daß
- für zumindest einen Teil der in einer Prozeß-Datei angegebenen Prozesse ein den Prozeß eindeutig kennzeichnender kryptographischer Wert gebildet wird,
- in der Prozeß-Datei jeweils der kryptographische Wert eines Prozesses enthalten ist,
- für den zugreifenden Prozeß dessen kryptographischer Wert gebildet wird, und
- bei der Überprüfung die kryptographischen Werte der Prozesse miteinander verglichen werden.

20. Anordnung nach einem der Ansprüche 17 bis 19,
bei der der Prozessor derart eingerichtet ist, daß in einem Aufrufmechanismus für eine Funktion eines Betriebssystem-

kerns, mit dem die Programme ausgeführt werden, ein Aufruf des zugreifenden Prozesses einer Überprüfungsfunktion zugeleitet wird, in der die Überprüfung erfolgt.

21. Anordnung nach einem der Ansprüche 17 bis 20, bei der der Prozessor derart eingerichtet ist, daß das Betriebssystem Windows NT ist.

22. Satz mehrerer Anordnungen und eine mit jeder Anordnung des Satzes mehrerer Anordnungen verbundenen Server-Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß,
wobei jede Anordnung einen Prozessor aufweist, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,
- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei wird für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, wird der zugreifende Prozeß gestartet oder weitergeführt, und
- sonst wird ein Alarmsignal generiert und an die Server-Anordnung gesendet,
und wobei die Server-Anordnung einen Prozessor aufweist, der derart eingerichtet ist, daß abhängig von mindestens einem empfangenen Alarmsignal eine vorgegebene Aktion ausgelöst wird.

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

09/763029

Absender: MIT DER INTERNATIONALEN VORLÄUFIGEN PRÜFUNG BEAUFTRAGTE BEHÖRDE

An:

SIEMENS AKTIENGESELLSCHAFT
Postfach 22 16 34
D-80506 München
ALLEMAGNE

PCT

MITTEILUNG ÜBER DIE ÜBERSENDUNG DES INTERNATIONALEN VORLÄUFIGEN PRÜFUNGSBERICHTS

(Regel 71.1 PCT)

Absendedatum *(Tag/Monat/Jahr)*	20.11.2000
Aktenzeichen des Anmelders oder Anwalts: GR 98P2369P	**WICHTIGE MITTEILUNG**

Internationales Aktenzeichen	Internationales Anmeldedatum *(Tag/Monat/Jahr)*	Prioritätsdatum *(Tag/Monat/Jahr)*
PCT/DE99/02013	01/07/1999	19/08/1998

Anmelder: SIEMENS AKTIENGESELLSCHAFT et al.

1. Dem Anmelder wird mitgeteilt, daß ihm die mit der internationalen vorläufigen Prüfung beauftragte Behörde hiermit den zu der internationalen Anmeldung erstellten internationalen vorläufigen Prüfungsbericht, gegebenenfalls mit den dazugehörigen Anlagen, übermittelt.

2. Eine Kopie des Berichts wird - gegebenenfalls mit den dazugehörigen Anlagen - dem Internationalen Büro zur Weiterleitung an alle ausgewählten Ämter übermittelt.

3. Auf Wunsch eines ausgewählten Amts wird das Internationale Büro eine Übersetzung des Berichts (jedoch nicht der Anlagen) ins Englische anfertigen und diesem Amt übermitteln.

4. **ERINNERUNG**

   Zum Eintritt in die nationale Phase hat der Anmelder vor jedem ausgewählten Amt innerhalb von 30 Monaten ab dem Prioritätsdatum (oder in manchen Ämtern noch später) bestimmte Handlungen (Einreichung von Übersetzungen und Entrichtung nationaler Gebühren) vorzunehmen (Artikel 39 (1)) (siehe auch die durch das Internationale Büro im Formblatt PCT/IB/301 übermittelte Information).

   Ist einem ausgewählten Amt eine Übersetzung der internationalen Anmeldung zu übermitteln, so muß diese Übersetzung auch Übersetzungen aller Anlagen zum internationalen vorläufigen Prüfungsbericht enthalten. Es ist Aufgabe des Anmelders, solche Übersetzungen anzufertigen und den betroffenen ausgewählten Ämtern direkt zuzuleiten.

   Weitere Einzelheiten zu den maßgebenden Fristen und Erfordernissen der ausgewählten Ämter sind Band II des PCT-Leitfadens für Anmelder zu entnehmen.

Name und Postanschrift der mit der internationalen Prüfung beauftragten Behörde	Bevollmächtigter Bediensteter
Europäisches Patentamt D-80298 München Tel. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465	Camps i Amigo, M.E. Tel. +49 89 2399-2237

NK

3T

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

PCT

INTERNATIONALER VORLÄUFIGER PRÜFUNGSBERICHT

(Artikel 36 und Regel 70 PCT)

Aktenzeichen des Anmelders oder Anwalts	**WEITERES VORGEHEN**	siehe Mitteilung über die Übersendung des internationalen vorläufigen Prüfungsbericht (Formblatt PCT/IPEA/416)
GR 98P2369P		
Internationales Aktenzeichen	Internationales Anmeldedatum *(Tag/Monat/Jahr)*	Prioritätsdatum *(Tag/Monat/Tag)*
PCT/DE99/02013	01/07/1999	19/08/1998
Internationale Patentklassifikation (IPK) oder nationale Klassifikation und IPK G06F9/46		
Anmelder SIEMENS AKTIENGESELLSCHAFT et al.		

1. Dieser internationale vorläufige Prüfungsbericht wurde von der mit der internationale vorläufigen Prüfung beauftragte Behörde erstellt und wird dem Anmelder gemäß Artikel 36 übermittelt.

2. Dieser BERICHT umfaßt insgesamt 9 Blätter einschließlich dieses Deckblatts.

   ☒ Außerdem liegen dem Bericht ANLAGEN bei; dabei handelt es sich um Blätter mit Beschreibungen, Ansprüchen und/oder Zeichnungen, die geändert wurden und diesem Bericht zugrunde liegen, und/oder Blätter mit vor dieser Behörde vorgenommenen Berichtigungen (siehe Regel 70.16 und Abschnitt 607 der Verwaltungsrichtlinien zum PCT).

   Diese Anlagen umfassen insgesamt 7 Blätter.

3. Dieser Bericht enthält Angaben zu folgenden Punkten:

   - I ☒ Grundlage des Berichts
   - II ☐ Priorität
   - III ☒ Keine Erstellung eines Gutachtens über Neuheit, erfinderische Tätigkeit und gewerbliche Anwendbarkeit
   - IV ☐ Mangelnde Einheitlichkeit der Erfindung
   - V ☒ Begründete Feststellung nach Artikel 35(2) hinsichtlich der Neuheit, der erfinderische Tätigkeit und der gewerbliche Anwendbarkeit; Unterlagen und Erklärungen zur Stützung dieser Feststellung
   - VI ☐ Bestimmte angeführte Unterlagen
   - VII ☒ Bestimmte Mängel der internationalen Anmeldung
   - VIII ☒ Bestimmte Bemerkungen zur internationalen Anmeldung

Datum der Einreichung des Antrags	Datum der Fertigstellung dieses Berichts
18/01/2000	20.11.2000
Name und Postanschrift der mit der internationalen vorläufigen Prüfung beauftragten Behörde: Europäisches Patentamt D-80298 München Tel. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465	Bevollmächtigter Bediensteter Schneider, M Tel. Nr. +49 89 2399 7509

INTERNATIONALER VORLÄUFIGER PRÜFUNGSBERICHT

Internationales Aktenzeichen PCT/DE99/02013

I. Grundlage des Berichts

1. Dieser Bericht wurde erstellt auf der Grundlage (*Ersatzblätter, die dem Anmeldeamt auf eine Aufforderung nach Artikel 14 hin vorgelegt wurden, gelten im Rahmen dieses Berichts als "ursprünglich eingereicht" und sind ihm nicht beigefügt, weil sie keine Änderungen enthalten.*):

Beschreibung, Seiten:

1-19 ursprüngliche Fassung

Patentansprüche, Nr.:

1-20 eingegangen am 29/09/2000 mit Schreiben vom 29/09/2000

Zeichnungen, Blätter:

1/6-6/6 ursprüngliche Fassung

2. Aufgrund der Änderungen sind folgende Unterlagen fortgefallen:

☐ Beschreibung, Seiten:
☒ Ansprüche, Nr.: 21, 22
☐ Zeichnungen, Blatt:

3. ☒ Dieser Bericht ist ohne Berücksichtigung (von einigen) der Änderungen erstellt worden, da diese aus den angegebenen Gründen nach Auffassung der Behörde über den Offenbarungsgehalt in der ursprünglich eingereichten Fassung hinausgehen (Regel 70.2(c)):

siehe Beiblatt

4. Etwaige zusätzliche Bemerkungen:

III. Keine Erstellung eines Gutachtens über Neuheit, erfinderische Tätigkeit und gewerbliche Anwendbarkeit

Folgende Teile der Anmeldung wurden nicht daraufhin geprüft, ob die beanspruchte Erfindung als neu, auf erfinderischer Tätigkeit beruhend (nicht offensichtlich) und gewerblich anwendbar anzusehen ist:

☐ die gesamte internationale Anmeldung.

☒ Ansprüche Nr. 16-20.

Begründung:

☐ Die gesamte internationale Anmeldung, bzw. die obengenannten Ansprüche Nr. beziehen sich auf den nachstehenden Gegenstand, für den keine internationale vorläufige Prüfung durchgeführt werden braucht (*genaue Angaben*):

☒ Die Beschreibung, die Ansprüche oder die Zeichnungen (*machen Sie hierzu nachstehend genaue Angaben*) oder die obengenannten Ansprüche Nr. 16-20 sind so unklar, daß kein sinnvolles Gutachten erstellt werden konnte (*genaue Angaben*):

siehe Beiblatt

☐ Die Ansprüche bzw. die obengenannten Ansprüche Nr. sind so unzureichend durch die Beschreibung gestützt, daß kein sinnvolles Gutachten erstellt werden konnte.

☐ Für die obengenannten Ansprüche Nr. wurde kein internationaler Recherchenbericht erstellt.

V. Begründete Feststellung nach Artikel 35(2) hinsichtlich der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit; Unterlagen und Erklärungen zur Stützung dieser Feststellung

1. Feststellung

Neuheit (N)	Ja:	Ansprüche	1-15
	Nein:	Ansprüche	
Erfinderische Tätigkeit (ET)	Ja:	Ansprüche	1-15
	Nein:	Ansprüche	
Gewerbliche Anwendbarkeit (GA)	Ja:	Ansprüche	1-15
	Nein:	Ansprüche	

2. Unterlagen und Erklärungen

siehe Beiblatt

VII. Bestimmte Mängel der internationalen Anmeldung

Es wurde festgestellt, daß die internationale Anmeldung nach Form oder Inhalt folgende Mängel aufweist:

siehe Beiblatt

VIII. Bestimmte Bemerkungen zur internationalen Anmeldung

Zur Klarheit der Patentansprüche, der Beschreibung und der Zeichnungen oder zu der Frage, ob die Ansprüche in vollem Umfang durch die Beschreibung gestützt werden, ist folgendes zu bemerken:

siehe Beiblatt

Es wird auf die folgenden Dokumente verwiesen:

D1: David Solomon, Inside Microsoft Windows NT, 2. Aufl., in deutscher Übersetzung erschienen in Microsoft Press 1998, englische Originalausgabe veröffentlicht im Mai 1998 bei Microsoft Press, Seiten 219-221, 303-323

D2: William Stallings, Operating Systems, 3. Aufl., erschienen bei Prentice Hall Dezember 1997, Seiten 632-636, 668,670-674.

Die Dokumente D1 und D2 wurden im internationalen Recherchenbericht nicht angegeben. Eine Kopie der Dokumente liegt bei.

Zu Punkt I
Grundlage des Berichts

Die nachfolgend aufgeführten Änderungen bringen Sachverhalte ein, die im Widerspruch zu Artikel 34 (2) b) PCT über den Offenbarungsgehalt der internationalen Anmeldung im Anmeldezeitpunkt hinausgehen:

(i) Die in den ursprünglichen unabhängigen Ansprüchen verwendete Formulierung

> "bei dem während des Ablaufs eines zu schützenden Programms und/oder einer zu schützenden Datei"

wurde in den geänderten unabhängigen Ansprüchen durchgängig durch die Formulierung

> "bei dem während des Ablaufs eines zu schützenden Programms und/oder einer Bearbeitung einer zu schützenden Datei"

ersetzt, ohne dass sich dafür eine Grundlage in der ursprünglichen Offenbarung findet.

(ii) Im geänderten Anspruch 4 wird das Merkmal neu eingeführt, dass

"die Überprüfungsfunktion als dynamisch bindbare Datei in den Adressraum des zu schützenden Programms und/oder der zu schützenden Datei eingebunden ist".

Aus der ursprünglichen Offenbarung geht zwar hervor, dass

(a) "jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils ein Adressraum zugeordnet ist" (s. ursprünglicher Anspruch 1 oder Beschreibung, S. 4, Z. 3-4), und dass
(b) "die Überprüfungsfunktion als dynamisch bindbare Datei in den Adressraum eingebunden ist" (s. ursprünglicher Anspruch 5).

Es ist jedoch für den Fachmann nicht ersichtlich, dass der in (b) genannte Adressraum sich auf den speziellen, in (a) genannten Adressraum bezieht und nicht auf den allgemeinen Adressraum eines Computersystems.

Deshalb wird der Bericht erstellt, als seien die Änderungen aus (i) und (ii) nicht gemacht worden.

Zu Punkt III
Keine Erstellung eines Gutachtens über Neuheit, erfinderische Tätigkeit und gewerbliche Anwendbarkeit

Eine Erstellung eines Gutachtens über Neuheit, erfinderische Tätigkeit und gewerbliche Anwendbarkeit ist für die Ansprüche 16-20 wegen Mangel an Klarheit (Artikel 6 PCT) aus folgenden Gründen nicht möglich:

Unter einem Prozessor wird in der Informatik im allgemeinen eine

"Funktionseinheit einer Rechenanlage, in der mindestens das Steuerwerk, das Rechenwerk und die zugehörigen Register zusammengefasst sind" (s. Duden Informatik, Bibliographisches Institut & Brockhaus AG, 1988)

verstanden.
Auch in der Anmeldung wird der Begriff Prozessor offenbar in diesem Sinne verwendet (s. Beschreibung, S. 8, Z. 11-13) .
Es ist nicht klar, wie die in den Ansprüchen 16, 17, 20 aufgezeigten Schritte

> eine Anordnung mit einem zur Durchführung dieser Schritte eingerichteten Prozessor

abgrenzen

> zu einer Anordnung mit einem bereits bekannten Prozessor,

da diese Schritte offenbar auch in einer Anordnung mit einem bereits bekannten Prozessor ohne eigene Anpassung z.B. des Steuerwerks, des Rechenwerks oder der Register, durchführbar sind.

Bemerkung:

Aus der vorliegenden Beschreibung (s. z.B. S. 18, Z. 21-25) wird klar, dass sich die Ansprüche 16-20 beziehen auf

> eine Anordnung in einem Computersystem zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozess,
> mit einem Prozessor, der die in Anspruch 17 bzw. 18 angegebenen Schritte durchführt,

bzw.

> einen Satz mehrerer Anordnungen in jeweils einem Computersystem und eine mit jeder Anordnung des Satzes mehrerer Anordnungen verbundenen Server-Anordnung in einem Computersystem zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozess,
> wobei jede Anordnung einen Prozessor aufweist, der die in Anspruch 20 für Prozessoren dieser Anordnungen angegebenen Schritte durchführt.

Ansprüche, die den Ansprüchen 16-20 entsprechen und zusätzlich derart klargestellt sind, sind hinsichtlich der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit genauso zu beurteilen wie die entsprechenden Verfahrensansprüche 1-3 und 6.

Zu Punkt V

Begründete Feststellung nach Artikel 35(2) hinsichtlich der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit; Unterlagen und Erklärungen zur Stützung dieser Feststellung

Das in den Ansprüchen 1-15 beschriebene Verfahren dient dem Schutz mehrerer Programme und/oder mehrerer Dateien vor einem unbefugten Zugriff durch einen Prozess.

Stand der Technik:

D1 und D2 offenbaren anhand des Betriebssystems Windows NT 4.0 ein Verfahren, das alle Merkmale des Gegenstandes der unabhängigen Ansprüche 1 und 2 beinhaltet, mit Ausnahme der Merkmale, die sich auf Bildung, Speicherung und den Vergleich kryptografischer Werte beziehen.

Problem:

Dem in den Ansprüchen 1-15 beschriebenen Verfahren liegt das Problem zu Grunde, mehrere Programme und/oder mehrere Dateien vor einem unbefugten Zugriff durch einen Prozess zu schützen unter Verwendung eines Betriebssystems, welches grundsätzlich Sicherheitslücken aufweist.

Lösung:

Das aus dem Stand der Technik bekannte Verfahren wird durch die in den Ansprüchen 1 und 2 angegebenen Schritte ergänzt, die sich auf Bildung, Speicherung und den Vergleich kryptografischer Werte beziehen.

THIS PAGE BLANK (USPTO)

Die Veränderung des Verfahrens aus dem Stand der Technik gemäß dieser Merkmale wird

- weder durch die im Recherchenbericht angegebenen Dokumente
- noch durch D1 oder D2
- noch durch das in Fachkreisen allgemein verbreitete Wissen, dass in einem Computersystem in einer Passwortdatei, durch die der Zugang zu Ressourcen, die den einzelnen Benutzer zugänglich gemacht sind, kontrolliert wird, die Passwörter der Benutzer als kryptografische Werte gespeichert werden,

aufgezeigt oder nahe gelegt.

Zu Punkt VII
Bestimmte Mängel der internationalen Anmeldung

(i) Die unabhängigen Ansprüche 1, 2, 16, 17, 20 sind nicht in der zweiteiligen Form nach Regel 6.3 b) PCT abgefaßt. Im vorliegenden Fall erscheint die Zweiteilung jedoch zweckmäßig. Folglich gehören die in Verbindung miteinander aus dem Stand der Technik bekannten Merkmale von Windows NT4.0, wie beschrieben in D1 und D2, in den Oberbegriff (Regel 6.3 b) i) PCT) und die übrigen Merkmale in den kennzeichnenden Teil (Regel 6.3 b) ii) PCT).

(ii) Die Merkmale der Ansprüche sind nicht mit in Klammern gesetzten Bezugszeichen versehen worden (Regel 6.2 b) PCT).

(iii) Die Beschreibung steht nicht, wie in Regel 5.1 a) iii) PCT vorgeschrieben, in Einklang mit den Ansprüchen.

(iv) Im Widerspruch zu den Erfordernissen der Regel 5.1 a) ii) PCT werden in der Beschreibung weder der in D1 und D2 bzgl. Windows NT4.0 offenbarte einschlägige Stand der Technik noch diese Dokumente angegeben.

Zu Punkt VIII

Bestimmte Bemerkungen zur internationalen Anmeldung

Siehe Bemerkungen zu Punkt III.

Weitere Bemerkungen:

Die Ansprüche 1-20 erfüllen nicht die Erfordernisse des Artikels 6 PCT.

Die Ansprüche 1 und 2 bzw. 16 und 17 wurden zwar als getrennte, unabhängige Ansprüche derselben Kategorie abgefaßt, sie scheinen sich aber tatsächlich bzgl. des zu schützenden Gegenstands zu überlappen und sind somit nicht knapp gefaßt. Dadurch ist es schwierig, den Gegenstand des Schutzbegehrens zu ermitteln, und Dritten wird die Feststellung des Schutzumfangs erschwert.

Patentansprüche

1. Verfahren zum Schutz mehrerer Programme und/oder mehrerer Dateien vor einem unbefugten Zugriff durch einen Prozeß,
- bei dem jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils ein Adreßraum zugeordnet ist,
- bei dem jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils eine Prozeß-Datei zugeordnet ist,
- bei dem in einer Prozeß-Datei gespeichert ist, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- bei dem für zumindest einen Teil der in einer Prozeß-Datei angegebenen Prozesse ein den Prozeß eindeutig kennzeichnender kryptographischer Wert gebildet wird,
- bei dem in der Prozeß-Datei jeweils der kryptographische Wert eines Prozesses enthalten ist,
- bei dem während des Ablaufs eines zu schützenden Programms und/oder einer Bearbeitung einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft wird, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- bei dem für den zugreifenden Prozeß dessen kryptographischer Wert gebildet wird,
- bei dem bei der Überprüfung die kryptographischen Werte der Prozesse miteinander verglichen werden
- bei dem für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, der zugreifende Prozeß gestartet wird, und
- sonst der zugreifende Prozeß nicht gestartet wird.

2. Verfahren zum Schutz mehrerer Programme und/oder mehrerer Dateien vor einem unbefugten Zugriff durch einen Prozeß,
- bei dem jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils ein Adreßraum zugeordnet ist,
- bei dem jedem zu schützenden Programm und/oder jeder zu schützenden Datei jeweils eine Prozeß-Datei zugeordnet ist,

- bei dem in einer Prozeß-Datei gespeichert ist, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- bei dem für zumindest einen Teil der in einer Prozeß-Datei angegebenen Prozesse ein den Prozeß eindeutig kennzeichnender kryptographischer Wert gebildet wird,
- bei dem in der Prozeß-Datei jeweils der kryptographische Wert eines Prozesses enthalten ist,
- bei dem während des Ablaufs eines zu schützenden Programms und/oder einer Bearbeitung einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft wird, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- bei dem für den zugreifenden Prozeß dessen kryptographischer Wert gebildet wird,
- bei dem bei der Überprüfung die kryptographischen Werte der Prozesse miteinander verglichen werden
- bei dem für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, der zugreifende Prozeß weitergeführt wird, und
- sonst der zugreifende Prozeß beendet wird.

3. Verfahren nach Anspruch 1 oder 3,
bei dem in einem Aufrufmechanismus für eine Funktion eines Betriebssystemkerns, mit dem die Programme ausgeführt werden, ein Aufruf des zugreifenden Prozesses einer Überprüfungsfunktion zugeleitet wird, in der die Überprüfung erfolgt.

4. Verfahren nach Anspruch 3,
bei dem die Überprüfungsfunktion als dynamisch bindbare Datei in den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei eingebunden wird.

5. Verfahren nach Anspruch 1 oder 2,

- bei dem ein Aufruf des zugreifenden Prozesses einer Überprüfungsfunktion zugeleitet wird, in der die Überprüfung erfolgt, und
- bei dem die Überprüfungsfunktion in einen Betriebssystemkern eines Betriebssystems, mit dem die Programme ausgeführt werden, integriert ist.

6. Verfahren nach einem der Ansprüche 1 bis 5,
bei dem das Betriebssystem Windows NT ist.

7. Verfahren nach einem der Ansprüche 1 bis 6,
bei dem in einem vorgebbaren Zeitabstand für jeden aktiven Prozeß, der zusammen mit einem zu schützenden Programm und/oder einer zu schützenden Datei abläuft, überprüft wird, ob der aktive Prozeß in der Prozeß-Datei, die dem zu schützenden Programm und/oder der zu schützenden Datei zugeordnet ist, enthalten ist und der Prozeß beendet wird, falls dies nicht der Fall ist.

8. Verfahren nach einem der Ansprüche 1 bis 7,
bei dem abhängig von einem vorgebbaren Ereignis für jeden aktiven Prozeß, der zusammen mit einem zu schützenden Programm und/oder einer zu schützenden Datei abläuft, überprüft wird, ob der aktive Prozeß in der Prozeß-Datei, die dem zu schützenden Programm zugeordnet ist, enthalten ist und der Prozeß beendet wird, falls dies nicht der Fall ist.

9. Verfahren nach einem der Ansprüche 1 bis 8,
bei dem ein Schutzprogramm zum Schutz des Verfahrens nach einem der vorangegangenen Ansprüche ausführbar ist, verschlüsselt gespeichert ist und zu Beginn des Verfahrens entschlüsselt wird

10. Verfahren nach einem der Ansprüche 1 bis 9,
bei dem die zu schützenden Programme und/oder die zu schützenden Dateien, verschlüsselt gespeichert sind und zu Beginn

des Verfahrens nach einem der vorangegangenen Ansprüche entschlüsselt werden.

11. Verfahren nach Anspruch 9,
bei dem nach der Entschlüsselung des Schutzprogramms dessen Integrität überprüft wird und das Verfahren nach einem der vorangegangenen Ansprüche nur ausgeführt wird, wenn die Integrität des Schutzprogramms gewährleistet ist.

12. Verfahren nach Anspruch 11,
bei dem nach der Integritätsprüfung des Schutzprogramms die Integrität aller in den Prozeß-Dateien enthaltenen Prozesse überprüft wird und das Verfahren nach einem der vorangegangenen Ansprüche nur ausgeführt wird, wenn die Integrität aller in den Prozeß-Dateien enthaltenen Prozesse gewährleistet ist.

13. Verfahren nach Anspruch 12,
bei dem nach der Integritätsprüfung der Prozesse die Integrität des zu schützenden Programms und/oder der zu schützenden Datei überprüft wird und das Verfahren nach einem der vorangegangenen Ansprüche nur ausgeführt wird, wenn die Integrität des zu schützenden Programms und/oder der zu schützenden Datei gewährleistet ist.

14. Verfahren nach Anspruch 12 oder 13,
bei dem mindestens eine der Integritätsprüfung unter Verwendung eines kryptographischen Verfahrens erfolgt.

15. Verfahren nach einem der Ansprüche 1 bis 14,
bei dem der kryptographische Wert durch Anwendung einer Hash-Funktion gebildet wird

16. Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß,
mit einem Prozessor, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:

- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,
- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- bei dem für zumindest einen Teil der in einer Prozeß-Datei angegebenen Prozesse ein den Prozeß eindeutig kennzeichnender kryptographischer Wert gebildet wird,
- bei dem in der Prozeß-Datei jeweils der kryptographische Wert eines Prozesses enthalten ist,
- bei dem während des Ablaufs eines zu schützenden Programms und/oder einer Bearbeitung einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft wird, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- bei dem für den zugreifenden Prozeß dessen kryptographischer Wert gebildet wird,
- bei dem bei der Überprüfung die kryptographischen Werte der Prozesse miteinander verglichen werden
- bei dem für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, der zugreifende Prozeß gestartet wird, und
- sonst der zugreifende Prozeß nicht gestartet wird.

17. Anordnung zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß,
mit einem Prozessor, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,

- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- bei dem für zumindest einen Teil der in einer Prozeß-Datei angegebenen Prozesse ein den Prozeß eindeutig kennzeichnender kryptographischer Wert gebildet wird,
- bei dem in der Prozeß-Datei jeweils der kryptographische Wert eines Prozesses enthalten ist,
- bei dem während des Ablaufs eines zu schützenden Programms und/oder einer Bearbeitung einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft wird, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- bei dem für den zugreifenden Prozeß dessen kryptographischer Wert gebildet wird,
- bei dem bei der Überprüfung die kryptographischen Werte der Prozesse miteinander verglichen werden
- bei dem für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, der zugreifende Prozeß weitergeführt wird, und
- sonst wird der zugreifende Prozeß beendet.

18. Anordnung nach Anspruch 16 oder 17,
bei der der Prozessor derart eingerichtet ist, daß in einem Aufrufmechanismus für eine Funktion eines Betriebssystemkerns, mit dem die Programme ausgeführt werden, ein Aufruf des zugreifenden Prozesses einer Überprüfungsfunktion zugeleitet wird, in der die Überprüfung erfolgt.

19. Anordnung nach einem der Ansprüche 16 bis 18,
bei der der Prozessor derart eingerichtet ist, daß das Betriebssystem Windows NT ist.

20. Satz mehrerer Anordnungen und eine mit jeder Anordnung des Satzes mehrerer Anordnungen verbundenen Server-Anordnung

zum Schutz mehrerer Programme vor einem unbefugten Zugriff durch einen Prozeß,
wobei jede Anordnung einen Prozessor aufweist, der derart eingerichtet ist, daß folgende Schritte durchführbar sind:
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils ein Adreßraum zugeordnet,
- jedem zu schützenden Programm und/oder jeder zu schützenden Datei ist jeweils eine Prozeß-Datei zugeordnet,
- in einer Prozeß-Datei ist gespeichert, welcher Prozeß oder welche Prozesse in dem jeweiligen Adreßraum ablaufen darf oder dürfen,
- bei dem für zumindest einen Teil der in einer Prozeß-Datei angegebenen Prozesse ein den Prozeß eindeutig kennzeichnender kryptographischer Wert gebildet wird,
- bei dem in der Prozeß-Datei jeweils der kryptographische Wert eines Prozesses enthalten ist,
- bei dem während des Ablaufs eines zu schützenden Programms und/oder einer Bearbeitung einer zu schützenden Datei für einen Prozeß, der auf den Adreßraum des zu schützenden Programms und/oder der zu schützenden Datei zugreifen will überprüft wird, ob der zugreifende Prozeß in der entsprechenden Prozeß-Datei angegeben ist,
- bei dem für den zugreifenden Prozeß dessen kryptographischer Wert gebildet wird,
- bei dem bei der Überprüfung die kryptographischen Werte der Prozesse miteinander verglichen werden
- bei dem für den Fall, daß der zugreifende Prozeß in der Prozeß-Datei angegeben ist, der zugreifende Prozeß gestartet oder weitergeführt wird, und
- sonst wird ein Alarmsignal generiert und an die Server-Anordnung gesendet,
und wobei die Server-Anordnung einen Prozessor aufweist, der derart eingerichtet ist, daß abhängig von mindestens einem empfangenen Alarmsignal eine vorgegebene Aktion ausgelöst wird.

09/763029

Translation

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference GR 98P2369P	FOR FURTHER ACTION	See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)
International application No. PCT/DE99/02013	International filing date (*day/month/year*) 01 July 1999 (01.07.99)	Priority date (*day/month/year*) 19 August 1998 (19.08.98)
International Patent Classification (IPC) or national classification and IPC G06F 9/46		
Applicant SIEMENS AKTIENGESELLSCHAFT		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 9 sheets, including this cover sheet.

☒ This report is also accompanied by ANNEXES, i.e., sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

These annexes consist of a total of 7 sheets.

3. This report contains indications relating to the following items:

- I ☒ Basis of the report
- II ☐ Priority
- III ☒ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- IV ☐ Lack of unity of invention
- V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- VI ☐ Certain documents cited
- VII ☒ Certain defects in the international application
- VIII ☒ Certain observations on the international application

Date of submission of the demand	Date of completion of this report
18 January 2000 (18.01.00)	20 November 2000 (20.11.2000)
Name and mailing address of the IPEA/EP Facsimile No.	Authorized officer Telephone No.

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/DE99/02013

I. Basis of the report

1. This report has been drawn on the basis of (*Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments.*):

☐ the international application as originally filed.

☒ the description, pages 1-19, as originally filed,
pages _____, filed with the demand,
pages _____, filed with the letter of _____,
pages _____, filed with the letter of _____.

☒ the claims, Nos. _____, as originally filed,
Nos. _____, as amended under Article 19,
Nos. _____, filed with the demand,
Nos. 1-20, filed with the letter of 29 September 2000 (29.09.2000),
Nos. _____, filed with the letter of _____.

☒ the drawings, sheets/fig 1/6-6/6, as originally filed,
sheets/fig _____, filed with the demand,
sheets/fig _____, filed with the letter of _____,
sheets/fig _____, filed with the letter of _____.

2. The amendments have resulted in the cancellation of:

☐ the description, pages _____
☒ the claims, Nos. 21.22
☐ the drawings, sheets/fig _____

3. ☒ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the Supplemental Box (Rule 70.2(c)).

4. Additional observations, if necessary:

See separate sheet.

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/DE99/02013

III. Non-establishment of opinion with regard to novelty, inventive step and industrial applicability

The questions whether the claimed invention appears to be novel, to involve an inventive step (to be non obvious), or to be industrially applicable have not been examined in respect of:

☐ the entire international application.

☒ claims Nos. 16-20

because:

☐ the said international application, or the said claims Nos. _____ relate to the following subject matter which does not require an international preliminary examination *(specify):*

☒ the description, claims or drawings *(indicate particular elements below)* or said claims Nos. 16-20 are so unclear that no meaningful opinion could be formed *(specify)*:

See separate sheet.

☐ the claims, or said claims Nos. _____ are so inadequately supported by the description that no meaningful opinion could be formed.

☐ no international search report has been established for said claims Nos. _____.

I. Basis of the report

1. This report has been drawn on the basis of *(Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments.)*:

CONTINUATION OF BOX I.3

The following amendments introduce substantive matter which goes beyond the disclosure of the international application as filed, thereby contravening PCT Article 34(2)(b):

(i) The wording used in the original independent claims,

"in which during the execution of a program and/or data file to be protected",

has been replaced in all amended independent claims by the wording,

"in which during the execution of a program to be protected and/or processing of a data file to be protected",

although no basis for this amendment can be found in the original disclosure.

(ii) In the amended Claim 4, a new feature is introduced, whereby

"the checking function is linked as a dynamically linkable file to the address space of the program and/or data file to be protected."

In the original disclosure, it is stated that

I. Basis of the report

1. This report has been drawn on the basis of *(Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments.)*:

(a) "an address space is associated with each program and/or data file to be protected" (see original Claim 1 or description, page 4, lines 3-4), and

(b) "the checking function is linked as a dynamically linkable data file to the address space" (see original Claim 5).

However, it is not clear to a person skilled in the art that the address space in point (b) concerns the special address space in point (a), and not the general address space of a computer system.

Consequently, the report is established as if the amendments (i) and (ii) had not been made.

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Continuation of: BOX III

An expert opinion on the novelty, inventive step and industrial applicability of Claims 16-20 is not possible for lack of clarity (PCT Article 6) regarding the following aspects:

In information technology, a processor is generally understood to be a

> "functional unit of a computer system combining at least the control unit, the arithmetic-logic unit and the associated register" (see *Duden Informatik*, Bibliographisches Institut & Brockhaus AG, 1988).

The application also clearly uses the term processor in this sense (see the description, page 8, lines 11-13).

It is not clear how the steps defined in Claims 16, 17 and 20 delimit a device for carrying out these steps from a device comprising the already known processor, since these steps can clearly also be carried out by a device comprising an already known processor, without any adaptation of the control unit, arithmetic-logic unit or register.

Observation:

It is clear from the present description (see e.g. page 18, lines 21-25) that Claims 16-20 concern

> a device in a computer system for protecting a plurality of programs from unauthorised access by a

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Continuation of: BOX III

process and comprising a processor that carries out the steps defined in Claims 17 and 18,

and

a set of a plurality of devices each in a computer system and a server arrangement connected to each device of the set of a plurality of devices in a computer system for protecting a plurality of programs from unauthorised access by a process, each device comprising a processor that carries out the steps defined in Claim 20 for the processors of said devices.

The novelty, inventive step and industrial applicability of claims corresponding to Claims 16-20 and clarified in this way should be assessed as those of the corresponding method Claims 1-3 and 6.

V. Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Statement

Novelty (N)	Claims	1-15	YES
	Claims		NO
Inventive step (IS)	Claims	1-15	YES
	Claims		NO
Industrial applicability (IA)	Claims	1-15	YES
	Claims		NO

2. Citations and explanations

This report makes reference to the following documents:

D1: David Solomon, Inside Microsoft Windows NT, second edition, German translation published by Microsoft Press 1998, English original edition published in May 1998 by Microsoft Press, pages 219-221, 303-323

D2: William Stallings, Operating Systems, third edition, published by Prentice Hall, December 1997, pages 632-636, 668, 670-674.

Documents D1 and D2 were not cited in the international search report. A copy of these documents is attached.

The method described in Claims 1-15 is used for protecting a plurality of programs and/or data files from unauthorised access by a process.

Prior art:

D1 and D2 disclose in the operating system Windows NT 4.0 a method having all the features of the subject matter of independent Claims 1 and 2, except for the features relating to the formation, storage and comparison of cryptographic values.

Problem:

The method described in Claims 1-15 addresses the problem of protecting a plurality of programs and/or data files from unauthorised access by a process, using an operating system suffering from fundamental security shortcomings.

Solution:

The method known from the prior art is complemented by the steps as per Claims 1 and 2, which concern the formation, storage and comparison of cryptographic values.

The alteration of the prior art method to include these features is neither disclosed nor suggested by

- the search report citations
- D1 or D2
- or the general professional knowledge that user passwords are stored as cryptographic values in a computer system having a password file for controlling access to resources made available to the individual users.

VII. Certain defects in the international application

The following defects in the form or contents of the international application have been noted:

(i) Independent Claims 1, 2, 16, 17 and 20 are not drafted in the two-part form defined by PCT Rule 6.3(b). However, the two-part form would appear to be appropriate in this case. Accordingly, the features of Windows NT 4.0 known in combination from the prior art (as described in documents D1 and D2) should be set out in a preamble (PCT Rule 6.3(b)(i)) and the remaining features should be specified in a characterising part (PCT Rule 6.3(b)(ii)).

(ii) The features of the claims are not followed by reference signs placed between parentheses (PCT Rule 6.2(b)).

(iii) Contrary to PCT Rule 5.1(a)(iii), the description is not in line with the claims.

(iv) Contrary to PCT Rule 5.1(a)(ii), the description does not cite documents D1 and D2 and does not indicate the relevant prior art concerning Windows NT 4.0 disclosed therein.

VIII. Certain observations on the international application

The following observations on the clarity of the claims, description, and drawings or on the question whether the claims are fully supported by the description, are made:

See observations in Box III.

Further observations:

Claims 1-20 do not meet the requirements of PCT Article 6.

Although Claims 1-2 and 16-17 have been drafted as separate independent claims of the same category, respectively, the subject matter for which protection is sought in these claims actually appears to overlap, and they are therefore not concise. This makes it hard to identify the subject matter for which protection is sought, and it is therefore unreasonably difficult for a third party to determine the scope of protection.

Foreign Version

Patent Claims

1. Method for protecting several programs and/or several files from unauthorized access by a process,

- in which each program and/or each file to be protected is assigned an address space,
- in which each program and/or each file to be protected is also assigned a process file,
- in which the process or processes that may run in the address space in question is or are stored in a process file,
- in which, during the running of a program and/or a file to be protected, for a process that attempts to access the address space of the program and/or the file to be protected, a check is made to confirm whether the accessing process is included in the corresponding process file,
- in which, if the accessing process is included in the process file, the accessing process is started, and
- otherwise, the accessing process is not started.

2. Method for protecting several programs and/or several files from unauthorized access by a process,

- in which each program and/or each file to be protected is assigned an address space,
- in which each program and/or each file to be protected is also assigned a process file,
- in which the process or processes that may run in the address space in question is or are stored in a process file,
- in which, during the running of a program and/or a file to be protected, for a process that attempts to access the address space of the program and/or a file to be protected, a check is made to confirm whether the accessing process is included in the corresponding process file,

Foreign Version

- in which, if the accessing process is included in the process file, the accessing process is continued, and

- otherwise, the accessing process is ended.

3. Method according to Claim 1 or 2,

- in which for at least a part of the processes included in a process file, a cryptographic value that uniquely identifies the process is formed,

- in which the cryptographic value of one process is contained in the process file,

- in which the accessing process's cryptographic value is formed, and

- in which the cryptographic values of the processes are compared with each other during the check.

4. Method according to one of the Claims 1 through 3,

in which in a call mechanism for a function of an operating system core with which the programs are executed, a call of the accessing process is forwarded to a checking function in which the check is carried out.

5. Method according to Claim 4,

in which the checking function can be integrated into the address space as a dynamically integrated file.

6. Method according to one of the Claims 1 through 3,

- in which a call of the accessing process is forwarded to a checking function in which the check takes place, and

- in which the checking function is integrated into an operating system core of an operating system with which the programs are executed.

7. Method according to one of the Claims 1 through 6,

in which the operating system is Windows NT.

Foreign Version

8. Method according to one of the Claims 1 through 7,

in which at a predetermined interval of time for each active process that runs along with a program and/or a file to be protected, a check is made to confirm whether the active process is contained in the process file that is assigned to the program and/or the file to be protected, and the process is ended if that is not the case.

9. Method according to one of the Claims 1 through 8,

in which in dependency on a predetermined event for each active process that runs along with a program and/or a file to be protected, a check is made to confirm whether the active process is contained in the process file that is assigned to the program to be protected, and the process is ended if that is not the case.

10. Method according to one of the Claims 1 through 9,

in which a protection program, which is set up in a way such that the method can be executed, is stored encoded and is decoded at the start of the method.

11. Method according to one of the Claims 1 through 10,

in which the programs and/or the files to be protected are stored encoded and are decoded at the start of the method.

12. Method according to Claim 10,

in which after the decoding of the protection program, its integrity is checked and the method is executed only if the integrity of the protection program is assured.

13. Method according to Claim 12,

in which after the integrity test of the protection program, the integrity of all processes contained in

the process files is checked and the method is executed only if the integrity of the protection program is assured.

14. Method according to Claim 13,

in which after the integrity test of the processes, the integrity of the program and/or the file to be protected is checked and the method is executed only if the integrity of the protection program is assured.

15. Method according to Claim 13 or 14,

in which at least one of the integrity tests takes place through the use of a cryptographic method.

16. Method according to one of the Claims 3 through 15,

in which the cryptographic value is formed through the use of a hash function.

17. Array for protecting several programs from unauthorized access by a process,

with a processor that is set up in a way such that the following steps can be carried out:

- an address space is assigned to each program and/or each file to be protected,
- a process file is assigned to each program and/or each file to be protected,
- the process or processes that may run in the address space in question is or are stored in a process file,
- during the running of a program and/or a file to be protected, for a process that attempts to access the address space of the program and/or the file to be protected, a check is made to confirm whether the accessing process is included in the corresponding process file,
- if the accessing process is included in the process file, the accessing process is started, and
- otherwise, the accessing process is not started.

18. Array for protecting several programs from unauthorized access by a process,

with a processor that is set up in a way such that the following steps can be carried out:

- an address space is assigned to each program and/or each file to be protected,

- a process file is assigned to each program and/or each file to be protected,

- the process or processes that may run in the address space in question is or are stored in a process file,

- during the running of a program and/or a file to be protected, for a process that attempts to access the address space of the program and/or the file to be protected, a check is made to confirm whether the accessing process is included in the corresponding process file,

- if the accessing process is included in the process file, the accessing process is continued, and

- otherwise, the accessing process is terminated.

19. Array according to Claim 17 or 18,

in which the processor is set up in such a way that

- for at least a part of the processes included in a process file, a cryptographic value that uniquely identifies the process is formed,

- the cryptographic value of one process is contained in the process file,

- for the accessing process whose cryptographic value is formed, and

- the cryptographic values of the processes are compared with each other during the check.

20. Array according to one of the Claims 17 through 19,

in which the processor is set up in such a way that in a call mechanism for a function of an operating

system core with which the programs are executed, a call of the accessing process is forwarded to a checking function in which the check is carried out.

21. Array according to one of the Claims 17 through 20,

in which the processor is set up in such a way that the operating system is Windows NT.

22. Set of several arrays and a server array which is connected with each array of the set of several arrays and which is to protect several programs from unauthorized access by a process,

whereby each array exhibits a processor that is set up in such a way that the following steps can be carried out:

- an address space is assigned to each program and/or each file to be protected,
- a process file is assigned to each program and/or each file to be protected,
- the process or processes that may run in the address space in question is or are stored in a process file,
- during the running of a program and/or a file to be protected, for a process that attempts to access the address space of the program and/or the file to be protected, a check is made to confirm whether the accessing process is included in the corresponding process file,
- if the accessing process is included in the process file, the accessing process is started or continued, and
- otherwise an alarm signal is generated and sent to the server array,

and whereby the server array exhibits a processor that is set up in such a way that a predetermined action is triggered in dependency on at least one received alarm signal.